April 2005
UILU-ENG-05-2206
DC-216

IMPROVING SUBSPACE SEGMENTATION WITH THE RAYLEIGH QUOTIENT

Shankar R. Rao, Andrew W. Wagner, Allen Y. Yang, and Yi Ma

Coordinated Science Laboratory
1308 West Main Street, Urbana, IL 61801
University of Illinois at Urbana-Champaign

REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. **REPORT DATE**	3. REPORT TYPE AND DATES COVERED
	April 8, 2005	

4. TITLE AND SUBTITLE
Improving Subspace Segmentation with the Rayleigh Quotient

5. FUNDING NUMBERS

6. **AUTHOR(S)**

Shankar R. Rao, Andrew W. Wagner, Allen Y. Yang, and Yi Ma

7. **PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**

Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
1308 West Main Street
Urbana, Illinois 61801-2307

8. **PERFORMING RGANIZATION REPORT NUMBER**

UILU-ENG-05-2206
(DC- 216)

9. **SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**

NSF

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. **SUPPLEMENTARY NOTES**

The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official position, policy or decision, unless so designated by other documentation

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

In this paper, we investigate the problem of subspace segmentation in the presence of significant noise. We draw upon multivariate discriminating statistics to improve the algebraic Generalized Principal Component Analysis method with the notion of Segmentation Polynomials*. A Segmentation Polynomial is a polynomial that both fits the data well and also provides the best segmentation of noisy samples drawn from different linear subspaces. We obtain Segmentation Polynomials by minimizing a ratio that is remarkably similar to the Rayleigh Quotient used in Fisher's Linear Discriminant. We will show that using a single Segmentation Polynomial, one can robustly segment data samples into linear subspaces in the presence of significant noise. We evaluate the performance of our method on highly noisy samples, and demonstrate its use in piece-wise linear fitting of nonlinear manifolds, segmenting color images, detection of straight lines in images, and sparse image representation.*

14. **SUBJECT TERMS**
subspace, polynomial, segmentation, clustering, GPCA, Rayleigh, robust, discriminant, Fisher

15. **NUMBER OF PAGES**
9 pages

16. PRICE CODE

17. **SECURITY CLASSIFICATION OF REPORT**	18. **SECURITY CLASSIFICATION OF THIS PAGE**	19. **SECURITY CLASSIFICATION OF ABSTRACT**	20. **LIMITATION OF ABSTRACT**
UNCLASSIFIED	**UNCLASSIFIED**	**UNCLASSIFIED**	**UL**

Improving Subspace Segmentation with the Rayleigh Quotient

Shankar R. Rao, Andrew W. Wagner, Allen Y. Yang, and Yi Ma
Coordinated Science Laboratory, University of Illinois
1308 W. Main, Urbana, Illinois 61801
Email: {yangyang,srrao,awagner,yima}@uiuc.edu

Abstract

In this paper, we investigate the problem of subspace segmentation in the presence of significant noise. We draw upon multivariate discriminating statistics to improve the algebraic Generalized Principal Component Analysis method with the notion of Segmentation Polynomials*. A Segmentation Polynomial is a polynomial that both fits the data well and also provides the best segmentation of noisy samples drawn from different linear subspaces. We obtain Segmentation Polynomials by minimizing a ratio that is remarkably similar to the Rayleigh Quotient used in Fisher's Linear Discriminant. We will show that using a single Segmentation Polynomial, one can robustly segment data samples into linear subspaces in the presence of significant noise. We evaluate the performance of our method on highly noisy samples, and demonstrate its use in piece-wise linear fitting of nonlinear manifolds, segmenting color images, detection of straight lines in images, and sparse image representation.*

1 Introduction

In this paper, we consider the problem of segmenting multiple subspaces in an ambient space from a set of noisy samples, sometimes referred to as the "subspace segmentation" problem. Typically, we assume that the dimensions of the subspaces are not known a priori, and that they can be different from one subspace to another. The solution to the special case of one subspace is the well known Principal Component Analysis (PCA) method [6], which has become one of the most important tools for data representation, approximation, and compression.

The general subspace segmentation problem is extremely important in computer vision since many practical applications are special cases of this common mathematical problem. For instance, due to the findings of [3, 7, 13, 17] it is now known that the multiple-motion segmentation problem is to a large extent a subspace segmentation problem, regardless of the camera model and motion model being considered. Subspace segmentation is also widely studied in pattern recognition since (linear) subspaces are good (first-order) approximations for many nonlinear data types such as human faces [19], textures [11], videos [5], and range data [14].

The general subspace segmentation problem itself is also extremely challenging due to the strong coupling between the problem of segmenting data into multiple subspaces and the problem of estimating the number/dimensions/bases of the subspaces. A large spectrum of effective solutions and algorithms have been developed and proposed for solving the subspace segmentation problem in the past few years [4, 9, 16, 18], thanks to the advancement of statistical learning methods such as K-means, Expectation Maximization (EM), Generalized Principal Component Analysis (GPCA), and RANSAC.

The GPCA method [18], provides an effective way to resolve the coupling between data segmentation and model estimation. Basically, it seeks a global algebraic representation of the unsegmented data set. The segmentation and estimation of individual subspaces can then be obtained by decomposing the global algebraic model into irreducible components, each component corresponding to exactly one subspace.

Although GPCA provides a theoretical solution to subspace segmentation, it does not yet answer some important questions regarding subspace segmentation with *noisy* data. The existing GPCA method relies on a linear least-square fitting to find the global algebraic model. Although such a fitting minimizes the modeling error, statistically it is not necessarily the best for the purpose of *segmentation* of the original data. In other words, among all of the polynomials that represent the data well, some of the polynomials may be far superior for giving an accurate and robust segmentation of subspaces from the noisy data, hence the name "Segmentation Polynomials."

In this paper, we will show how to incorporate the Rayleigh quotient (Section 2) into GPCA (Section 3) to identify Segmentation Polynomials and subsequently improve the performance of the existing GPCA method (Section 4). Besides the theoretical justification, we will verify the improvement by showing how the generalized eigenvalue spectrum of the Rayleigh quotient may improve the spectrum of the polynomials that fit the data. We also show how the data can be denoised by projecting onto the zero set of the Segmentation Polynomial and hence improve the performance

of the existing GPCA algorithm with noisy data. We validate our method with many simulations (Section 5) and real applications (Section 6).

2 Review of the Fisher Discriminant

Fisher's Linear Discriminant improves upon PCA for the purpose of discrimination by using labeled training samples. We would like to apply a similar idea to GPCA. That is, we would like to find a polynomial that is more optimal for subspace segmentation. In order to understand how the techniques derived in the Fisher Discriminant may be applied to subspace segmentation, we first review the key ideas of the Fisher Discriminant via a simple example. We will show in Section 3 that for GPCA, the derivative of the Veronese map plays the same role as the class labels in the Fisher Discriminant.

For simplicity, we consider the case for finding the best line to discriminate between samples drawn from two clusters when the data samples are projected onto the line. Without loss of generality, we can consider a line to be a one-dimensional subspace, and so it can be specified by a single basis vector $\boldsymbol{c} \in \mathbb{R}^K$. Thus for each data sample $\boldsymbol{x} \in \mathbb{R}^K$, the projection of that sample onto the line is simply $\hat{\boldsymbol{x}} = \boldsymbol{c}^T\boldsymbol{x} = \boldsymbol{x}^T\boldsymbol{c} \in \mathbb{R}$.

2.1 Minimizing the Within-Cluster Distance

Ideally, we would like all data samples to be as close to their respective cluster means as possible. Thus for a given cluster, we can derive the following metric to minimize.

$$\begin{aligned} J_A &\doteq \sum_{n=1}^{N_i} \|\hat{\boldsymbol{x}}_n - \hat{\boldsymbol{\mu}}_i\|^2 = \sum_{n=1}^{N_i} \|\boldsymbol{x}_n^T\boldsymbol{c} - \boldsymbol{\mu}_i^T\boldsymbol{c}\|^2 \\ &= \boldsymbol{c}^T\Big(\sum_{n=1}^{N_i}(\boldsymbol{x}_n - \boldsymbol{\mu}_i)(\boldsymbol{x}_n - \boldsymbol{\mu}_i)^T\Big)\boldsymbol{c} \doteq \boldsymbol{c}^T A_i \boldsymbol{c} \end{aligned}$$

with $\{\boldsymbol{x}_n\}_{n=1}^{N_i}$ belong to the ith cluster, $i = 1, 2$. We will call A_i the *within-class scatter matrix* for the ith cluster:

Definition 1 (Within-Class Scatter Matrix)**.** *The* within-class scatter matrix A_i *for a subset of* N_i *data samples with the cluster mean* $\boldsymbol{\mu}_i$ *is defined as*

$$A_i \doteq \sum_{n=1}^{N_i}(\boldsymbol{x}_n - \boldsymbol{\mu}_i)(\boldsymbol{x}_n - \boldsymbol{\mu}_i)^T \quad \in \mathbb{R}^{K\times K}. \tag{1}$$

The total within-class scatter matrix for the two clusters is $A \doteq A_1 + A_2$.

2.2 Maximizing the Between-Cluster Distance

We would also like the cluster means themselves to be as far apart as possible:

$$\begin{aligned} J_B &\doteq \|\hat{\boldsymbol{\mu}}_1 - \hat{\boldsymbol{\mu}}_2\|^2 = \|\boldsymbol{\mu}_1^T\boldsymbol{c} - \boldsymbol{\mu}_2^T\boldsymbol{c}\|^2 \\ &= \boldsymbol{c}^T(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T\boldsymbol{c} \doteq \boldsymbol{c}^T B\boldsymbol{c}. \end{aligned}$$

Definition 2 (Between-Class Scatter Matrix)**.** *The* between-class scatter matrix B *between two clusters with cluster means* $\boldsymbol{\mu}_1$*, and* $\boldsymbol{\mu}_2$ *is defined to be*

$$B \doteq (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T \quad \in \mathbb{R}^{K\times K}. \tag{2}$$

2.3 The Rayleigh Quotient

In order to find the line $\boldsymbol{c}$ that simultaneously minimizes the projected within class scatter $\boldsymbol{c}^T A\boldsymbol{c}$ and maximizes the projected between class scatter $\boldsymbol{c}^T B\boldsymbol{c}$, we can minimize the ratio of these two measurements. This ratio is known as the *Rayleigh Quotient*:

Definition 3 (Rayleigh Quotient)**.** *For two square symmetric matrices* $A, B \in \mathbb{R}^{K\times K}$ *and a vector* $\boldsymbol{c} \in \mathbb{R}^K$*, the* Rayleigh Quotient R *is the ratio*

$$R \doteq \frac{\boldsymbol{c}^T A\boldsymbol{c}}{\boldsymbol{c}^T B\boldsymbol{c}}. \tag{3}$$

It can be shown that the unit vector $\boldsymbol{c}$ that minimizes the Rayleigh Quotient is the minimal generalized eigenvector of the matrix pair (A, B). Recall that any generalized eigenvector $\boldsymbol{c}$ of (A, B) satisfies

$$A\boldsymbol{c} = \lambda B\boldsymbol{c} \quad \text{for some} \quad \lambda \in \mathbb{R}.$$

Thus, if B is invertible, then $\boldsymbol{c}$ is just the eigenvector of the matrix $B^{-1}A$ associated with the smallest eigenvalue $\lambda_{\min}$.

3 Rayleigh Quotient and GPCA

The basic idea of GPCA, as described in [18], is to fit the entire data set sampled from a mixture of subspaces with a set of polynomials so that the subspaces are the common zero set of the polynomials. Then each subspace corresponds to a "factor" of these polynomials.[1]

Using the Veronese map, also known in machine learning as the polynomial embedding, data samples drawn from a mixture of linear subspaces will lie on a single linear subspace when lifted into the much higher dimensional feature space. Thus, to fit polynomials to the data set, we must find coefficients $\{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_n\}$ that satisfy

$$p_i(\boldsymbol{x}) = \boldsymbol{c}_i^T\nu(\boldsymbol{x}) = 0, \quad i = 1, \ldots, n, \tag{4}$$

for all $\boldsymbol{x}$ in the data set.

[1] Strictly speaking, the subspaces correspond to the prime ideals of the primary decomposition of the ideal generated by these polynomials.

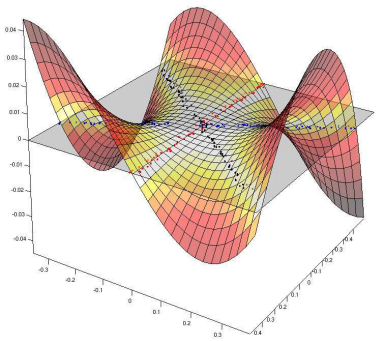

Figure 1: A polynomial in two variables whose zero set is three lines in $\mathbb{R}^2$.

If the data samples are drawn from a mixture of hyperplanes, then we have $n = 1$ and the polynomial $p(\boldsymbol{x})$ is factorable:

$$p(\boldsymbol{x}) = \prod_{j=1}^{G} (\boldsymbol{b}_j^T \boldsymbol{x}) \tag{5}$$

with G the number of (different) hyperplanes and $\boldsymbol{b}_j$ the normal vector to the jth plane. Figure **??** shows such a polynomial which fits three lines in $\mathbb{R}^2$. For a single polynomial, it is very easy to find the coefficient vector $\boldsymbol{c}$. The kernel of the space spanned by the vectors $\nu(\boldsymbol{x})$ is only one-dimensional, so the singular value decomposition will readily yield this vector up to a scale. Once we have $\boldsymbol{c}$, the normal vectors of the hyperplanes are readily obtained by taking the gradient of the polynomial $p(\boldsymbol{x})$ at the data points.

However, if the data samples are drawn from a mixture of linear subspaces, not all of which are hyperplanes in $\mathbb{R}^K$, then $n > 1$, and the vectors $\boldsymbol{c}_i$ will be linear combinations of the coefficients of those factorable polynomials (5).

3.1 Minimizing the Error of the Fitted Polynomial

Let us begin by examining a factorable polynomial $p(\boldsymbol{x})$ whose zero set contains the subspaces $\{S_j\}_{j=1}^{G}$:

$$p(\boldsymbol{x}) = \prod_{j=1}^{G} (\boldsymbol{b}_j^T \boldsymbol{x}) = \boldsymbol{c}^T \nu(\boldsymbol{x}) = 0,\ \forall \boldsymbol{x} \in S = \bigcup_{j=1}^{G} S_j. \tag{6}$$

In the presence of noise, it is likely that $p(\boldsymbol{x}) \neq 0$, but we would like to find the unit coefficient vector $\boldsymbol{c}$ that minimizes the following fitting error

$$J_A \doteq \sum_{n=1}^{N} |p(\boldsymbol{x}_n)|^2 = \sum_{n=1}^{N} \boldsymbol{c}^T \nu(\boldsymbol{x}_n) \nu(\boldsymbol{x}_n)^T \boldsymbol{c} \tag{7}$$

$$= \boldsymbol{c}^T \Big(\sum_{n=1}^{N} \nu(\boldsymbol{x}_n) \nu(\boldsymbol{x}_n)^T \Big) \boldsymbol{c} = \boldsymbol{c}^T A \boldsymbol{c}. \tag{8}$$

J_A is also known as the algebraic distance. Compared to the J_A of discriminant analysis, here we do not need to know the "class" information, i.e., which subspace each $\boldsymbol{x}_n$ belongs to. From the definition of $p(\boldsymbol{x})$, $|p(\boldsymbol{x}_n)|^2$ is small as long as it belongs to one of the subspaces.
We will call A the *within-subspace scatter matrix*:

Definition 4 (Within-Subspace Scatter Matrix)**.** *The* within-subspace scatter matrix *for a set of N data samples is given by*

$$A \doteq \sum_{n=1}^{N} \nu(\boldsymbol{x}_n) \nu(\boldsymbol{x}_n)^T. \tag{9}$$

The eigenvectors of A associated with the smallest eigenvalues form a basis for the coefficients $\boldsymbol{c}_i$ of all the polynomials that fit the data set with any given error threshold. Nevertheless, the polynomial that minimizes J_A is not necessarily the best for separating the noisy data into their respective subspaces.

3.2 Maximizing the Distance from Other Subspaces

Let us examine the derivative of the polynomial at each of the data samples. Let us assume that the data sample $\boldsymbol{x}_1$ lies exclusively in the subspace S_1. Then we have:

$$\nabla_{\boldsymbol{x}} p(\boldsymbol{x}_1) = \Big(\prod_{j=2}^{G} \boldsymbol{b}_j^T \boldsymbol{x}_1 \Big) \boldsymbol{b}_1 = \boldsymbol{c}^T \nabla_{\boldsymbol{x}} \nu(\boldsymbol{x}_1) \tag{10}$$

as all the other terms containing $\boldsymbol{b}_1^T \boldsymbol{x}_1$ are zero and hence dropped out. The direction of $\nabla_{\boldsymbol{x}} p(\boldsymbol{x}_1)$ in (10) is the same as the vector $\boldsymbol{b}_1$, and its magnitude is given by

$$\|\nabla_{\boldsymbol{x}} p(\boldsymbol{x}_1)\| = \Big| \Big(\prod_{j=2}^{G} \boldsymbol{b}_j^T \boldsymbol{x}_1 \Big) \Big|. \tag{11}$$

That is, $\|\nabla_{\boldsymbol{x}} p(\boldsymbol{x}_1)\|$ is the "product of the distances" of $\boldsymbol{x}_1$ to all the subspaces that it does not belong to.
We have assumed $\boldsymbol{x}_1$ to be exclusively in S_1, and consequently, all of the terms of the product in (11) will be nonzero, and in fact, the further $\boldsymbol{x}_1$ is from $\bigcup_{j=2}^{G} S_j$, the union of the other subspaces in the model, the larger $\|\nabla_{\boldsymbol{x}} p(\boldsymbol{x}_1)\|$ will be. This is a highly desirable trait for the segmentation purpose, so we would like to find the unit coefficient vector $\boldsymbol{c}$ that maximizes $\|\nabla_{\boldsymbol{x}} p(\boldsymbol{x})\|$.

$$J_B \doteq \sum_{n=1}^{N} \|\nabla_{\boldsymbol{x}} p(\boldsymbol{x}_n)\|^2 \tag{12}$$

$$= \boldsymbol{c}^T \Big(\sum_{n=1}^{N} \nabla_{\boldsymbol{x}} \nu(\boldsymbol{x}_n) \nabla_{\boldsymbol{x}} \nu(\boldsymbol{x}_n)^T \Big) \boldsymbol{c} = \boldsymbol{c}^T B \boldsymbol{c}.$$

We will call B the *between-subspace scatter matrix*:

Definition 5 (Between-Subspace Scatter Matrix)**.** *The* between-subspace scatter matrix *for a set of N data samples is given by*

$$B \doteq \sum_{n=1}^{N} \nabla_{\boldsymbol{x}} \nu(\boldsymbol{x}_n) \nabla_{\boldsymbol{x}} \nu(\boldsymbol{x}_n)^T. \tag{13}$$

3.3 The Rayleigh Quotient for Subspaces

The coefficient vector that simultaneously minimizes the polynomial evaluated at each of the samples while maximizing the norm of the derivative at each point is obtained by simply minimizing the ratio of these two metrics.

Definition 6 (Segmentation Polynomial)**.** *The* Segmentation Polynomial $p(\boldsymbol{x}) = \boldsymbol{c}^T \nu(\boldsymbol{x})$ *is specified by the coefficient vector $\boldsymbol{c}^*$ such that*

$$\boldsymbol{c}^* = \arg\min_{\boldsymbol{c}} \frac{\boldsymbol{c}^T A \boldsymbol{c}}{\boldsymbol{c}^T B \boldsymbol{c}}. \tag{14}$$

This ratio looks remarkably similar to the Rayleigh quotient[2] described earlier in Fisher Discriminant. The minimization of the Rayleigh quotient only requires that A and B are real, symmetric, positive semidefinite matrices. Thus the vector $\boldsymbol{c}^*$ that minimizes this ratio will be the minimal generalized eigenvector of A and B. In our context, the within-subspace scatter matrix B will be full rank, because otherwise all of the data samples can be fitted with polynomials of degree lower than G. As a result, the vector $\boldsymbol{c}^*$ is simply the eigenvector of $B^{-1}A$ associated with the smallest eigenvalue. The normalization of A by B^{-1} significantly improves the singular-value spectrum of A: It makes the null space of A much more robust to the corruption of noise, which makes the estimation of all polynomials that fit the data a better-conditioned problem. We will illustrate this with a quantitative example in Section 4.2 (see Figure **??**).

Rayleigh Quotient for Clusters v.s. for Subspaces. Let us compare the within-class scatter matrix (Def. 1) and the within-subspace scatter matrix (Def. 4). The former measures the squared Euclidean distance between samples and their cluster means; the latter measures the square of the fitting polynomial evaluated at each sample, which can be regarded as a squared "algebraic distance" between samples and the linear subspaces they lie on. Similarly, we compare the between-class scatter matrix (Def. 2) and the between-subspace scatter matrix (Def. 5). The former measures the squared Euclidean distance between cluster means; the latter measures the squared norm of the derivative of the polynomial evaluated at each sample, which can be regarded as a squared "distance" between samples and all of the other linear subspaces in the model.

[2]Beware not to confuse the ratio $(\sum \|p(\boldsymbol{x})\|^2)/(\sum \|\nabla p(\boldsymbol{x})\|^2)$ with the Sampson distance $\sum(\|p(\boldsymbol{x})\|^2/\|\nabla p(\boldsymbol{x})\|^2)$ commonly used in polynomial fitting.

4 Improving GPCA with the Segmentation Polynomial

4.1 Projecting Noisy Data onto the Segmentation Polynomial

In the presence of noise, the data samples will not be exactly in the zero set of the Segmentation Polynomial used to fit the data, $\{\boldsymbol{x} : p(\boldsymbol{x}) = \boldsymbol{c}^T \nu(\boldsymbol{x}) = 0\}$. Thus, to reduce noise, for every data sample $\boldsymbol{x}$, we would ideally like to find the closest point on the zero set of the polynomial, that is,

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{y}} |\boldsymbol{x} - \boldsymbol{y}|^2, \text{ subject to } \boldsymbol{c}^T \nu(\boldsymbol{y}) = 0. \tag{15}$$

Unfortunately, in general, there is no closed-form solution to this problem. However, observe that the steepest descent direction of the original data sample $\boldsymbol{x}$ to the zero set of the polynomial is given by the derivative

$$\boldsymbol{n} \doteq \nabla_{\boldsymbol{x}} p(\boldsymbol{x}) = \boldsymbol{c}^T \nabla_{\boldsymbol{x}} \nu(\boldsymbol{x}). \tag{16}$$

Then a good approximate solution to the projection $\hat{\boldsymbol{x}} = \boldsymbol{x} + \alpha \boldsymbol{n}$ where $\alpha \in \mathbb{R}$ is the smallest real root of the polynomial equation

$$p(\boldsymbol{x} + \alpha \boldsymbol{n}) = \boldsymbol{c}^T \nu(\boldsymbol{x} + \alpha \boldsymbol{n}) = 0. \tag{17}$$

Figure 2 demonstrates the effectiveness of projecting noisy data samples onto the zero set of their Segmentation Polynomial. These particular data samples were drawn from two lines L_1 and L_2 and a plane P in $\mathbb{R}^3$, with 5% additive Gaussian noise. The original data samples (blue), are connected to the projected data samples (red) by black lines.

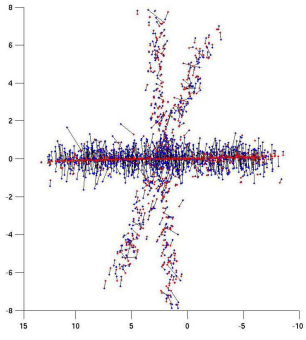

Figure 2: Results of projecting noisy data samples onto the zero set of their Segmentation Polynomial.

The projection is a simple and efficient method to improve the fidelity of the data samples. There are fitting polynomials for which this method will not improve segmentation (see Figure 3). Also note that it is possible to use gradient-descent techniques in this way, but such methods will often require many iterations. We recognize that there are many closest point approximation algorithms in the literature, but this is not a main focus of this paper.

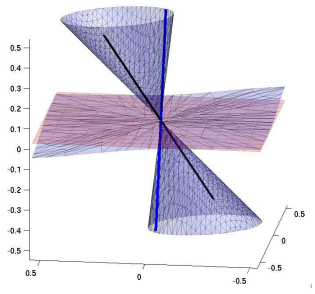

Figure 3: The zero set of a Segmentation Polynomial that is not factorable into linear subspaces.

4.2 Clustering Normals

Once we have computed a robust Segmentation Polynomial and projected the data samples onto this Segmentation Polynomial, we cluster the normals to the polynomial at each point into groups. The normals for projected data samples from a single subspace will form a cluster, and so any reasonable technique that will cluster a set of samples into a given number of groups will be sufficient for segmentation. We have decided to use a modified version of the K-means algorithm. The classical K-means algorithm groups the data into zero-dimensional "clusters," with each cluster specified by its cluster mean. For our version, we instead use the SVD of the set of normals, to compute the prototype normal for each subspace.[3]

We believe that clustering these normals is a far more robust way of segmenting the data than the method of finding representative points for each subspace, proposed by [18], because segmentation does not rely on the fidelity of individual samples. Figure **??** exhibits the clustering of the normals of the Segmentation Polynomial for the data samples drawn from two lines and a plane in $\mathbb{R}^3$.

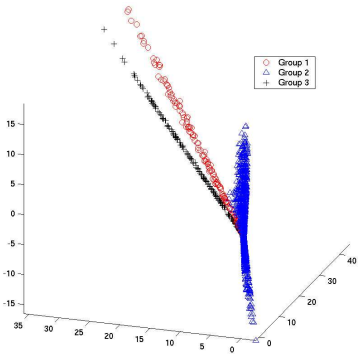

Figure 4: Clustering results for the normals by K-means.

[3]In order to robustly estimate the dimension of each subspace, we first use groups of two normals to remove data samples that are on the intersection of the hyperplanes specified by the normals. These data samples are not used for estimation of the individual linear subspaces.

It is natural to ponder what happens if we use all the polynomials that fit the data for segmentation. In fact, the Rayleigh quotient makes the estimation of all such polynomials more robust to noise. To see this, let us consider the same data set drawn from the two lines and one plane in $\mathbb{R}^3$. As Figure **??** illustrates, the generalized eigenvalues of the Rayleigh Quotient provide a much sharper "knee point" than the singular values of A, with which we can more easily estimate the number of polynomials that fit the data (in this case four polynomials).[4]

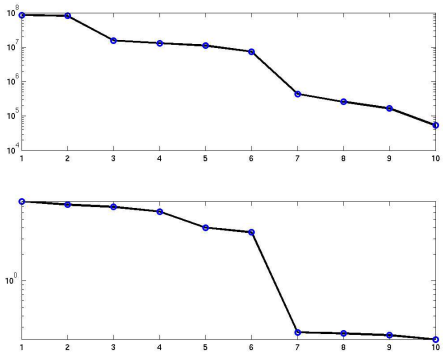

Figure 5: Top: Plot of the singular values of the within-subspace scatter matrix A. Bottom: Plot of the eigenvalues of the matrix $B^{-1}A$ derived from the Rayleigh quotient.

However, the derivatives of all the polynomials give us more than one normal at each sample. Clustering these groups of normals robustly is, to a large extent, a dual problem to the original subspace segmentation problem, that may not be much simpler. Thus for multiple Segmentation Polynomials, we are left with finding one representative point per subspace and using the orthogonal complement of the space spanned by these normals as a basis for each subspace. As stated in Section 4.2, we believe that clustering of the normals provided by the most dominant Segmentation Polynomial is far more robust in the presence of noise.

We provide below an outline of the overall Rayleigh Quotient-based GPCA algorithm.

4.3 Comparisons with other methods

For our first experiment, we generate data samples from five linear subspaces in $\mathbb{R}^5$, with dimension 1, 2, 2, 3, and 4 respectively, and then add 2% Gaussian noise. This is a very challenging subspace segmentation example despite the seemingly moderate noise because of the large number and varying dimensions of the subspaces. We ran data sets of this dimension configuration 20 times each for K-Subspaces[5],

[4]Thus, one could significantly improve the robustness of the GPCA algorithm proposed in [18] by simply thresholding the generalized eigenvalues of the Rayleigh Quotient instead.

[5]For our experiment, K-Subspaces is given the above dimension configuration, and initializes with random basis vectors for each subspace.

Algorithm 1 Rayleigh Quotient-Based GPCA Algorithm.

Given: a set of data samples $\boldsymbol{X} = \{\boldsymbol{x}\}$, and the number of subspaces G,

for all $\boldsymbol{x} \in \boldsymbol{X}$ **do**
  Compute the Veronese map $\nu(\boldsymbol{x})$ of order G and its derivative at $\boldsymbol{x}$.
end for
$A \leftarrow \sum_{\boldsymbol{x} \in \boldsymbol{X}} \nu(\boldsymbol{x})\nu(\boldsymbol{x})^T$.
$B \leftarrow \sum_{\boldsymbol{x} \in \boldsymbol{X}} \nabla_{\boldsymbol{x}}\nu(\boldsymbol{x})\nabla_{\boldsymbol{x}}\nu(\boldsymbol{x})^T$.
$\boldsymbol{c} \leftarrow \arg\min_{\hat{\boldsymbol{c}}} \frac{\hat{\boldsymbol{c}}^T A \hat{\boldsymbol{c}}}{\hat{\boldsymbol{c}}^T B \hat{\boldsymbol{c}}}$ = the eigenvector corresponding to the smallest eigenvalue $\lambda_{\min}$ of $B^{-1}A$.
for all $\boldsymbol{x} \in \boldsymbol{X}$ **do**
  $\boldsymbol{n} \leftarrow \boldsymbol{c}^T \nabla_{\boldsymbol{x}}\nu(\boldsymbol{x})$.
  $\alpha \leftarrow \min\{\text{real roots of } \boldsymbol{c}^T\nu(\boldsymbol{x} + \hat{\alpha}\boldsymbol{n})\}$.
  $\hat{\boldsymbol{x}} \leftarrow \boldsymbol{x} + \alpha\boldsymbol{n}$.
  $\hat{\boldsymbol{n}} \leftarrow \boldsymbol{c}^T \nabla_{\boldsymbol{x}}\nu(\hat{\boldsymbol{x}})$.
end for
Group the normals (and hence the data samples) into G groups using the K-means algorithm.
For each group of samples, use PCA to determine the dimension and basis vectors for their subspace.

the prior GPCA algorithm, our Rayleigh-Quotient based GPCA algorithm, and our algorithm as an initialization for the K-Subspaces algorithm. The results of the experiment are detailed below. The experiment verifies that the particular problem is challenging. In fact the only reasonable performance is occurs when our GPCA algorithm is used as an initialization for K-Subspaces.

Method	Misclassification Rate
K-Subspaces	47.97%
Existing GPCA	39.09%
Our GPCA	35.10%
Our GPCA + K-Subspaces	14.99%

Robustness Testing Figure **??** shows an example data set that we will use to fully character the difference in performance between our algorithm and the existing GPCA method. Notice in this case that the zero set of the dominant Segmentation Polynomial in general (asymptotically) approximates a union of three planes, which results in a correct segmentation of the three subspaces – the two lines are contained in two of of the planes, respectively.

For our experiment, we draw 1000 samples from the plane $z = 0$ and 200 samples from each of two lines. These lines are constructed with random angles that ensure a minimum of 30° subspace angle difference. We then add between 1% and 7% Gaussian noise and apply both the existing GPCA algorithm and our algorithm to the data set, instructing them to search for three linear subspaces. This test was performed 1000 times at each noise level, and for each test run the misclassification rate was computed using the known a priori sample labels.

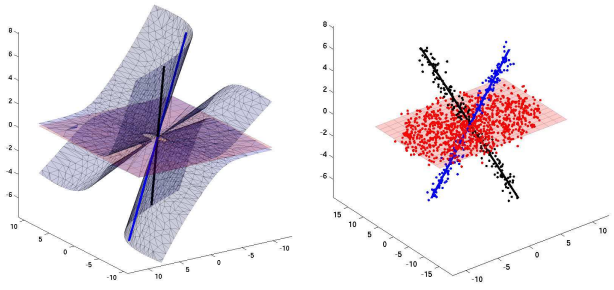

Figure 6: Left: The zero set of a Segmentation Polynomial for data samples drawn from two lines and a plane with 5% additive Gaussian noise. Right: The set of subspaces estimated by our algorithm.

Figure **??** shows the result of our experiment. The average misclassification rate is displayed as a function of the noise level. These results verify that while the two algorithms

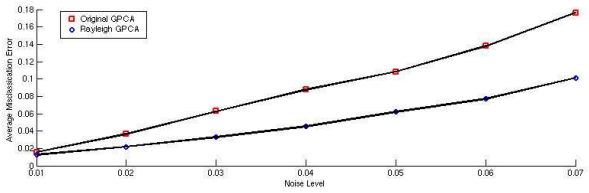

Figure 7: Plot of average misclassification error as a function of the noise for the existing GPCA algorithm and our algorithm.

have a negligible difference in error for low noise level, as we increase the amount of noise, the difference in performance becomes much more dramatic. The ability of any set of subspaces to segment noisy data samples becomes limited as noise increases since samples near the intersections of the subspaces are more likely mis-classified. The results show that our algorithm approaches this limit.

To better understand the performance of our algorithm, we analyze the distribution of misclassification rates over the 1000 test runs for a given noise level. In Figure 8, the misclassifications rates for 1000 test runs of the data set with 6% noise are sorted and displayed as a distribution. These distributions reveal that both algorithms have performance types that belong to one of three categories: (Class A), where the model and the segmentation are estimated correctly; (Class B), where the segmentation is reasonable, but the model estimation is incorrect (i.e., one or more of the subspaces has incorrect dimension); and (Class C), where neither the model nor the segmentation is correct. As Figure 8 demonstrates, even in the presence of 6% noise, our algorithm produces a meaningful segmentation of the noisy data samples almost 98% of the time.

Though GPCA is designed to simultaneously model and segment mixtures of linear subspaces, it can be used to provide a piecewise linear approximation to non-linear struc-

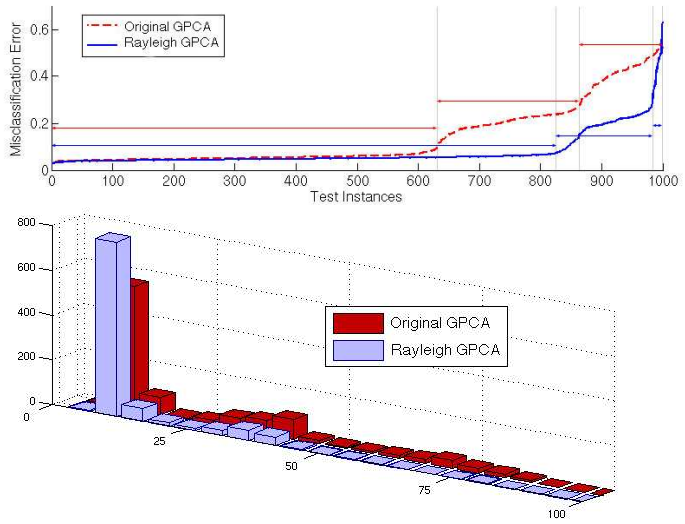

Figure 8: Top: Plot of sorted misclassification error for 1000 test runs of GPCA and our algorithm for 6% additive Gaussian noise. Bottom: Histogram of the above misclassification error. The horizontal axis is percent misclassification error and the vertical axis the number of trials with a given misclassificaton error.

tures as well. As most real world data is not strictly piecewise linear, the nonlinearity can be treated as noise. Then the ability of our algorithm dealing with noisy data will provide strong justification for its potential success in many real world applications. For some previous work in this area, please refer to [12, 15].

Figure 9 shows an example with data samples drawn from a hemisphere in $\mathbb{R}^3$ with radius 5 in the presence of 2% Gaussian noise. We instruct our algorithm to fit this non-linear data set with three and six subspaces, respectively. The results of segmentation and subspaces fitted to the data set are displayed in Figure 9. Notice that the segmentation results for six groups resemble half of a dodecahedron, a regular polyhedron with twelve congruent planar faces. The results suggest that the algorithm places the six planes almost evenly around the sphere.

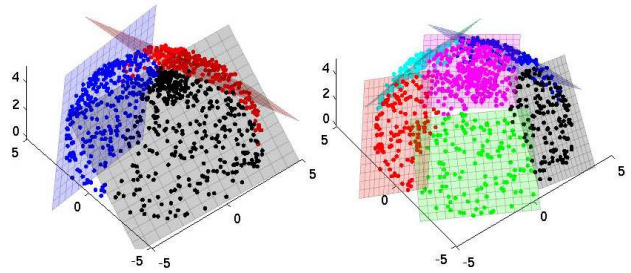

Figure 9: Planes fitted to samples drawn from a hemisphere in $\mathbb{R}^3$ clustered by our algorithm into three and six groups, respectively.

5 Experiments and Applications

In this section we illustrate the potential of our algorithm with several applications to traditional image processing tasks, namely color segmentation, line fitting, and sparse image representation. We do not claim that GPCA will provide an optimal solution for any of these problems, but instead demonstrate that our Rayleigh Quotient-based GPCA algorithm can be applied very easily to many different areas.

5.1 Color-Based Image Segmentation

In this experiment, we use our GPCA algorithm to adaptively choose a linear model for each distinctly colored object in a scene. The distribution of the pixel color values of the two images of Figure **??** (top) in the RGB color space has been rendered in Figure 10. As we see, certain objects form fairly distinct lines in the color space (e.g. the grey from the fence posts) while others form draping surfaces (e.g. the red-orange from the tomatoes).

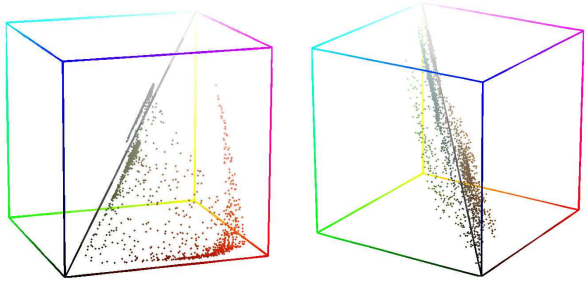

Figure 10: The distribution in the RGB color space of the images shown at the top of Figure **??**.

We embed the three normalized color intensities for each pixel as a vector in $\mathbb{R}^4$ using homogeneous coordinates $[R, G, B, 1]^T$. Our algorithm currently handles around two thousand data samples, so we sub-sample the data before passing it into the algorithm. The resulting subspaces are used to segment the full data set. The segmentation results for the two images are shown in Figure **??**. The results are reasonable given the simple embedding, but do not compare to results from single-purpose algorithms, such as [8].

5.2 Global Line Detection and Fitting

Traditional methods for detecting line features usually detect local edges, using algorithms such as the Canny edge detector [1] and the Mean Shift algorithm [2], and then pass the local edges to a global line-fitting algorithm, such as connected component analysis. In this experiment, we present a very simple method for globally detecting and fitting lines in an image using our algorithm, without any local edge detection.

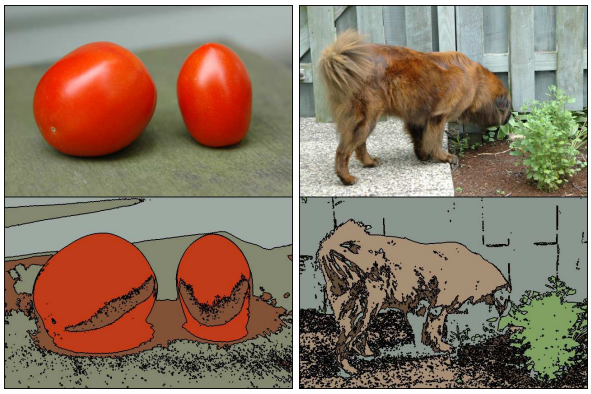

Figure 11: Color-based segmentation of two natural images.

For every image $I(x, y)$, we embed every pixel into a three-dimensional vector $\boldsymbol{v}(x, y) \doteq \|\nabla I(x, y)\|\big[x, y, 1\big]^T \in \mathbb{R}^3$. The norm of the image gradient modulates the likelihood of the pixel being an edge pixel.[6] If a collection of pixels $\{(x, y)\}$ in the image form a line, they satisfy the equation $ax + by + c = 0$. Or equivalently, the vectors $\{\boldsymbol{v}(x, y)\}$ associated with these pixels lie on a plane in $\mathbb{R}^3$. Thus, by applying our algorithm to fit a number of 2-D planes to all the vectors, we globally detect and fit a number of lines – the normal vector $[a, b, c]^T$ of every plane gives the parameters of a line in the image.

We apply this simple line fitting method to two natural images, with the results shown in Figure **??**. Note how the algorithm "hallucinates", detecting global lines that are interrupted by an arbitrarily large gap, as evidenced by the gaps between the triangles and the occlusion of the sidewalk by the pickup truck. Detected edges are of varying width, as seen in the crosswalk stripes, and need not be perfectly straight, as seen both in the crosswalk stripe and in the intentionally misaligned triangles. Except for the number of groups and the data itself, the parameters passed to our algorithm are identical.

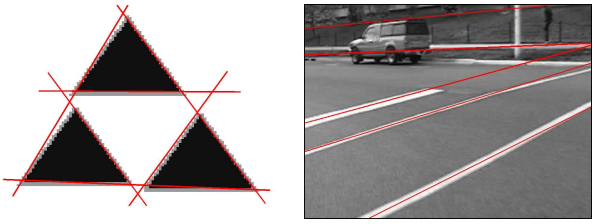

Figure 12: Detection and Segmentation of straight lines in a synthetic and a natural image.

PCA is a popular method of finding a lower-dimensional unimodal sparse representation of images. By using GPCA, we can estimate a mixture of linear subspaces, with each subspace modeling a different texture. That makes the model applicable to ensembles of images. In our database, we select two groups of 8 gray-scale images from the Berkeley segmentation dataset [10] shown in Figure 13. One group contains common natural scenes, and the other contains more structured urban scenes. We randomly sample 100 blocks of 8 by 8 windows from each image,[7] stack them into 64-dimensional vectors, and apply our algorithm to obtain a hybrid linear model.

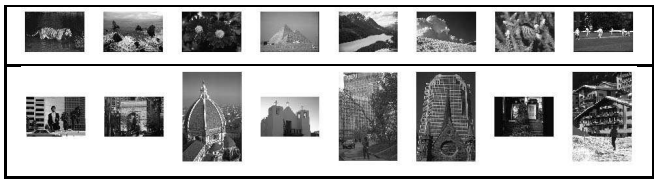

Figure 13: Top: natural scenes. Down: Urban scenes.

In this example, we preset the desired subspace number as 3, and the algorithm identifies the bases for the subspaces for each group. Figure **??** shows all basis vectors as 8 by 8 windows. Visually, these vectors capture the essential difference between natural scenes and urban scenes.

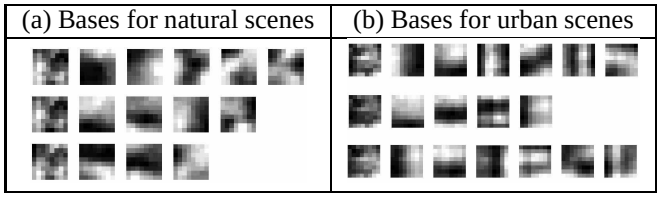

Figure 14: Results showing basis vectors of segmented subspaces.

In this paper we have provided both theoretical and empirical verification that the performance of the existing generalized PCA method is significantly improved by incorporating it with a Rayleigh quotient for subspaces. In the future, we will examine model selection criteria to robustly determine the number of groups.

References

[1] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, November 1986.

[2] D. Comanicu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:603–619, May 2002.

[3] J. Costeira and T. Kanade. A multibody factorization method for independently moving objects. In *International Journal on Computer Vision*, pages 159–179, 1998.

[6] This is the basis for most local edge detection methods.

[7] The window size is limited by computer memory.

[4] J. Ho, M. H. Yang, J. Lim, K.C. Lee, and D. Kriegman. Clustering appearances of objects under varying illumination conditions. In *Proceedings of International Conference on Computer Vision and Pattern Recognition*, 2003.

[5] K. Huang, Y. Ma, and R. Vidal. Minimum effective dimension for mixtures of subspaces: a robust GPCA algorithm and its applications. In *Proceedings of International Conference on Computer Vision and Pattern Recognition*, volume 2, pages 631–638, 2004.

[6] I. Jolliffe. *Principal component analysis*. Springer, 1986.

[7] Y. Kanazawa and K. Kanatani. Do we really have to consider covariance matrices for image features? In *Proceedings of IEEE International Conference on Computer Vision*, pages 301–306, 2001.

[8] G. Klinker, S.A. Shafer and T. Kanade. Color Image Analysis With An Intrinsic Reflection Model In *Proceedings of IEEE International Conference on Computer Vision*, pages 292–296, 1988.

[9] A. Leonardis, H. Bischof, and J. Maver. Multiple eigenspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (35):2613–2627, 2002.

[10] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

[11] R. Ramamoorthi. Analytic PCA construction for theoretical analysis of lighting variability, including attached shadows, in a single image of a convex lambertian object. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1322–1333, 2002.

[12] S. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *SCIENCE*, 290:2323–2326, 2000.

[13] P. Saisan, G. Doretto, Y. Wu, and S. Soatto. Dynamic texture recognition. In *Proceedings of International Conference on Computer Vision and Pattern Recognition*, number 2, pages 58–63, 2001.

[14] G. Taubin. Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 13(11):1115–1138, 1991.

[15] J. Tenenbaum, Vin de Silva, and J. C. Langford. A global geometric framework for nonlinear demensionality reduction. *SCIENCE*, 290:2319–2323, 2000.

[16] M. Tipping and C. Bishop. Mixture of probabilistic principal component analyzers. *Neural Computation*, 11(2), 1999.

[17] R. Vidal and Y. Ma. A unified algebraic approach to 2-D and 3-D motion segmentation. In *Proceedings of European Conference on Computer Vision*, 2004.

[18] R. Vidal, Y. Ma, and J. Piazzi. A new GPCA algorithm for clustering subspaces by fitting, differentiating and dividing polynomials. In *Proceedings of International Conference on Computer Vision and Pattern Recognition*, 2004.

[19] M. Yang, N. Ahuja, and D. Kriegman. Face recognition using kernel eigenfaces. In *Proceedings of IEEE International Conference on Image Processing*, number 1, pages 37–40, 2000.